Amendments to Specification:

Please replace the paragraph beginning on page 6, line 20 with the following amended paragraph.

> In one aspect of the present invention, the configuration server can be an LDAP server. In another aspect of the present invention, the active application components can be instances of [[Java]] JAVA programming language classes (henceforth Java classes) classes. In yet another aspect of the present invention, the configuration client can include a notifier object and a listener interface. In consequence, the active application components can be configured to receive update notifications from the configuration client through the listener interface. Similarly, the configuration server also can include a notifier object and a listener interface. Likewise, the configuration client can be configured to receive update notifications from the configuration server through the listener interface.

Please replace the paragraph beginning on page 11, line 6 with the following amended paragraph.

> As shown in Figure 2, the configuration client 202 can be managed separately from the platform 203. Moreover, in one aspect of the invention, the configuration client 202 can be implemented as a [[Java Bean]] JAVA BEAN software object. In consequence, the configuration client 202 [[Java Bean]] JAVA BEAN software object implementation can be reused by other applications. Notably, the platform 203 can be a [[Java]] JAVA programming language-based platform for managing application components 205. As such, the platform 203 can behave as an object request broker (ORB) for the various application components 205 installed on the platform 203.

Please replace the paragraph beginning on page 11, line 13 with the following amended paragraph.

> In operation, when the client position 207 undergoes bootstrap, the platform 203 can start the configuration client 202. Once started, the configuration client 202 can read a bootstrap properties file containing only the uniform resource locator (URL) of the configuration server 201 and a client position identifier. Subsequently, the configuration client 202 can

establish a communications connection to the configuration server 201. In particular, the communications connection can be an LDAP connection to an LDAP server using [[the]] a [[Java]] JAVA Naming and Directory Interface (JNDI/LDAP). Still, the invention is not limited in regard to the particular communications connection. Rather, any suitable communications connection can suffice. For example, the communications connection can be a point-to-point TCP/IP connection to a CGI interface to a flat-file database using HTTP.

Please replace the paragraph beginning on page 13, line 19 with the following amended paragraph.

Once the updated configuration data and software modules have been retrieved, the configuration client 202 can notify the platform 203 that updates have been received. For example, in a [[Java or C++]] JAVA programming language or a C++ programming language implementation, the configuration client 202 can generate an update received event and transmit the same to the platform 203. In response, if updated software modules have been received, the platform 203 can identify application components to which the updates are directed. Where an update is updated configuration data which does not require the termination of the targeted application component 205, the platform 203 can simply deliver the update to the targeted application component 205. Specifically, the configuration client 202 can notify the targeted application component 205 that updated configuration data has been received. In response, the targeted application components 205 can retrieve the updated configuration data from the configuration client 202 and can reinitialize associated internal states appropriately.